THOMSON - BRANDT

DPT : ORDINATEUR FAMILIAL

N° : 1-4-63

Rev : 0

CLASSIFICATION : Document de travail

DATE : 18 April 1986

TACHE : Machine 86

SOUS-TACHE : Logiciel

RELECTURE :

AUTEUR : PA

TITRE : Presentation de CGI

MOTS CLES : CGI, CGM ,VDI ,VDC

REFERENCES : CGI : Second Initial Draft, November 1985, ISO (WG2 N356)
CGM : Transfer and Storage of Picture Description Information, ISO/DIS 8632

RESUME :

CGI est une proposition de standard graphique, interfacant le code independant et le code dependant du materiel utilise.

CGI donne acces a un peripherique virtuel, dont l'espace de coordonnees est suppose infini, et dont les fonctionnalites sont uniformisees. CGI propose un ensemble de fonctions de controle, et de fonctions graphiques, et prend en charge les mappings et ajustements a realiser pour visualiser l'image virtuelle sur un terminal donne.

CGM offre un moyen de transport des images graphiques, en proposant une construction standard de fichiers graphiques, dans lesquels sont stockees les images virtuelles definies sous CGI.

REVISION :

FAITE PAR :

LE :

DESTINATAIRES :

VD, HH, JRM ,BM, PP ,PA, ER, TS, ZW

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

CGI : Computer Graphics Interface

1 PRESENTATION

1.1 Definition de CGI

CGI est une proposition de standard graphique, construite sur la notion de terminal virtuel, se situant dans la hierarchie des couches de logiciel, a un niveau tres interne, proche du systeme.

Interface utilisateur :

CGI offre aux applications (ou aux Workstations de systemes graphiques de plus haut niveau comme GKS), une interface avec un peripherique graphique virtuel (VDI : Virtual Device Interface), independant de toute contrainte physique. CGI propose un protocole de dialogue avec ce terminal virtuel, et prend en charge l'interpretation des commandes graphiques a destination d'un terminal reel quelconque.

Interface systeme :

CGI interface les drivers des peripheriques disponibles. Ces drivers peuvent etre reduits au minimum si le peripherique pilote repond a un protocole CGI (cas de quelques imprimantes).

1.2 Structure et conformite d'une implementation

CGI distingue 3 classes de peripheriques virtuels, chaque classe impliquant un noyau minimum specifique :

- OUTPUT : fonctions de sortie seulement ; noyau minimum defini par CGIMIN (40 fonctions environ, parmi 140 proposees).

- INPUT : fonctions d'entree seulement ; noyau minimum defini par INPUT CGIMIN (40 fonctions, parmi une centaine).

- OUTIN : fonctions d'E/S ; noyau minimum = CGIMIN + INPUT CGIMIN.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

Toute implementation qui se veut conforme au draft propose doit comporter :

- un noyau minimum, correspondant a la classe de peripheriques supportes.

- des fonctions optionnelles, qui permettent d'elargir CGI de facon assez consequente. Il est a noter que se limiter au noyau minimum presente peu d'interet, l'economie d'un certain nombre de fonctions n'etant pas justifiable en regard de la souplesse d'utilisation apportee.

- eventuellement un ou plusieurs "OPTION SETS" : ce sont des ensembles de routines repondant a une fonctionnalite commune (gestion d'objets, fonctions raster ...). Chaque ensemble est lui meme constitue d'un noyau minimum (obligatoirement supporte si l'ensemble est declare present) et de fonctions non obligatoires.

Toutes les fonctions implantees doivent repondre aux specifications CGI (tout au moins pour leur interface et leur comportement, leur nom mnemonique n'etant pas fourni par CGI)

Enfin, CGI offre 2 possibilites pour implanter des fonctions non standardisees :

- ESCAPE pour les fonctions de controle (fait partie du noyau minimum).

- GDP (Generalized Drawing Primitive) pour les fonctions de dessin (routine hors noyau minimum).

A noter que CGI dispose de toute une série de fonctions optionnelles, permettant a une application d'interroger le terminal sur ses possibilites, pour utiliser au mieux les ressources graphiques disponibles.

2 PLAN DU DOCUMENT

Ce document est une description generale de CGI, et plus precisement de OUTPUT CGI. Les fonctions d'entree ne seront donc pas abordees ici. Cette presentation etant faite en vue d'une implementation sur la machine 16 bits, les extensions de CGI rendues necessaires par l'environnement VSDD seront egalement decrites.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

La premiere partie definit l'environnement de travail d'une session CGI : les espaces de coordonnees utilises (virtuel et reel) et les attributs de controle du mapping effectue : comment conserver des "pixels carres" ou au contraire imposer des etirements ou compressions ?, effets de miroirs ...

Ensuite les possibilites graphiques offertes par CGI sont presentees : fonctions de dessin et attributs associes (types de lignes, remplissages, patterns.....)

Une troisieme partie presente quelques fonctionnalites supportees par CGI, mais non imposees dans un noyau minimum :

- segmentation : identification d'une liste de fonctions comme une seule entite (un segment).
- groupes d'attributs : memorisation de listes de valeurs pour les attributs de dessin, correspondant a des configurations frequemment utilisees.
- les figures fermees quelconques : permettent de relier implicitement des lignes, arcs et polylines, et de remplir les figures ainsi construites.
- gestion de bitmaps.

Enfin une breve description de CGM (Computer Graphics Metafile) sera donnee :

- presentation.
- utilisation parallele a CGI.
- codages des fichiers images (codage binaire, caractere et texte)

En conclusion, un resume des limites rencontrees, et des lacunes de CGI sera presente, suivi de la liste des fonctions probablement implementees sur la machine 16 bits.

3 ENVIRONNEMENT CGI

3.1 VDC space

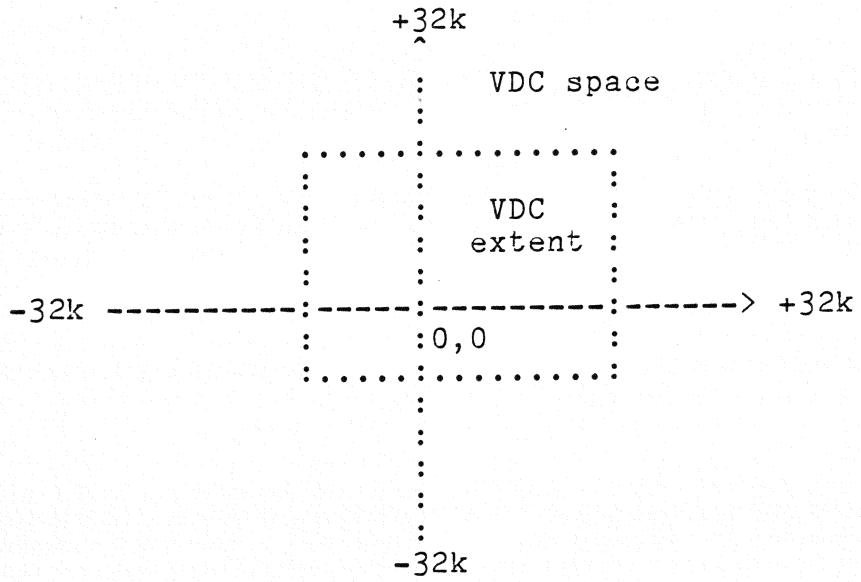
L'espace virtuel de travail de CGI est un espace cartesien, muni d'un systeme de coordonnees orthonorme, en theorie de dimension et de precision infinies, appele "VDC space" (Virtual Device Coordinates). Une implementation de CGI fixera l'environnement virtuel, le minimum a supporter etant l'utilisation de coordonnees entieres sur 16 bits (espace de -32k a +32k en x et en y).

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

Note : CGI prevoit mais n'impose pas l'utilisation de coordonnees reelles.

3.2 VDC extent

Un sous ensemble de l'espace virtuel est ensuite construit, en specifiant 2 points de VDC space qui definissent la diagonale d'un rectangle parallele aux axes x et y. VDC extent represente le fenetre virtuelle qui sera mappee sur l'espace reel : tout point de VDC extent sera associe a un point de la fenetre de visualisation, et tout point exterieur a VDC extent sera exterieur a cette fenetre.



3.3 Espace physique

3.3.1 Cas general

C'est le terminal de sortie. Deux notions sont a differencier : le systeme de coordonnees propre au terminal, lui permettant d'adresser des unites elementaires de visualisation (des pixels pour un moniteur), defini par un point origine et les limites en x et en y, et les dimensions métriques de la surface disponible specifiees en dizièmes de mm.

CGI connait les caractéristiques physiques de la surface de visualisation utilisee lors d'une session (notamment sa resolution et ses dimensions au 1/10 mm pres). L'application peut decider de n'utiliser qu'une partie de cette surface ; VDC extent sera alors mappe sur la fenetre (et non pas sur toute la surface disponible). Cette fenetre est une zone rectangulaire, appelee "viewport",

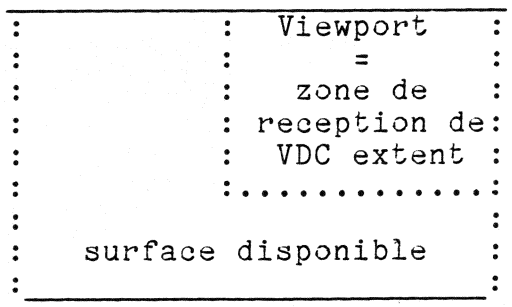
Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

definie par 2 coins opposes. Il y a 3 facons d'exprimer ces points:

- en pourcentage de la surface disponible (unites prises par defaut par CGI) : par exemple l'utilisation de la demi page superieure d'une imprimante donnerait (0,0) et (100,50), pour une origine fixee en haut, a gauche. Quel que soit le terminal utilise, ces unites garantissent un rapport constant entre la taille du dessin et la taille de la surface disponible.

- en 1/10 de mm : ces unites permettent de conserver , ou de fixer la dimension d'un dessin, quelle que soit la surface disponible.

- en coordonnees propres au terminal : cette solution est la plus precise mais elle implique la connaissance des caracteristiques du terminal.



3.3.2 Application a un environnement VSDD

Si la notion de "surface disponible" et de "surface de visualisation" est simple a concevoir dans le cas d'une imprimante, elle est plus complexe a cerner dans le cas d'un environnement VSDD : la surface disponible est-elle definie par l'ecran ..., par la fenetre ou est visualise l'objet ..., ou par le bitmap associe a l'objet VSDD ??

Le VSDD gerant lui-meme le positionnement de l'objet sur l'ecran, et utilisant un systeme de coordonnees locales pour chaque objet, il est logique de faire travailler CGI sur ce meme systeme de coordonnees.

-----> la surface disponible sera le bitmap associe a l'objet. Ses caracteristiques physiques decoulent des caracteristiques de l'ecran et du systeme de coordonnees locales utilise :

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- origine en haut a gauche.
- dimensions logiques = hauteur et largeur du bitmap en pixels.
- dimensions physiques : calculees par CGI d'apres les dimensions de l'ecran et sa definition.

-----> le viewport sera une partie du bitmap, dans lequel sera mappe VDC extent (c'est un premier niveau de clipping).

3.4 Controle du Mapping

Les 2 espaces concernes sont VDC extent et le viewport.

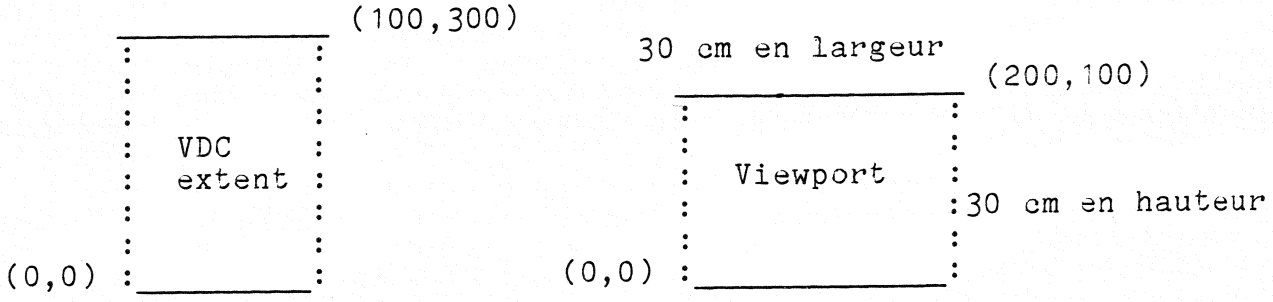
3.4.1 Isotropie

C'est la conservation des rapports de distances dy/dx, dans l'espace virtuel d'une part et dans l'espace metrique reel d'autre part. Autrement dit, un cercle defini dans VDC space restera un cercle quelle que soit la dimension des pixels du terminal utilise. Cette contrainte peut cependant etre ignoree, CGI offrant 2 modes de fonctionnement :

- Mapping libre :

VDC extent est mappe sur la totalite du viewport, sans tenir compte des dimensions physiques.

exemple : VDC defini par (0,0) et (100,300)
Viewport defini par (0,0) et (200,100)
Largeur = 30 cm et Hauteur = 30 cm.
(cas de pixels 2 fois plus hauts que larges)



Mapping defini par CGI :
x = 2*x et y = y/3
Rapport hauteur/largeur dans VDC = 3

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

Rapport hauteur/largeur dans viewport (metrique) = 1

---> tous les dessins vont etre aplatis d'un tiers !!!!
Un cercle de rayon 10 dans VDC va devenir une ellipse de rayons 60 et 10 dans le viewport (9cm et 3 cm).

- Isotropie forcee :

La conservation des rapports dy/dx est assuree en reduisant le viewport sur une des 2 dimensions (jamais les 2), afin d'assurer un mapping isotropique. CGI redefinit un viewport effectif :

Dans notre exemple, il faut garder le rapport de 3 specifie pour VDC extent. Le viewport effectif va donc mesurer 10 cm sur 30, et sa definition en coordonnees (0,0) et (67,100).

Nouveau Mapping : x = 0.66 * x et y = y/3

Notre cercle de rayon 30 devient une ellipse de rayons 20 et 10 dans le viewport effectif (3 cm et 3 cm donc un cercle !)

Note : le controle du mapping permet a l'utilisateur de choisir entre ces 2 modes et dans le cas d'un mapping isotropique, de decider de quelle facon sera reduit le viewport : pour une coupure verticale, le viewport effectif pourra etre cadre a gauche, au centre ou a droite du viewport initialement defini (respectivement en haut, au centre ou a droite pour une coupure horizontale).

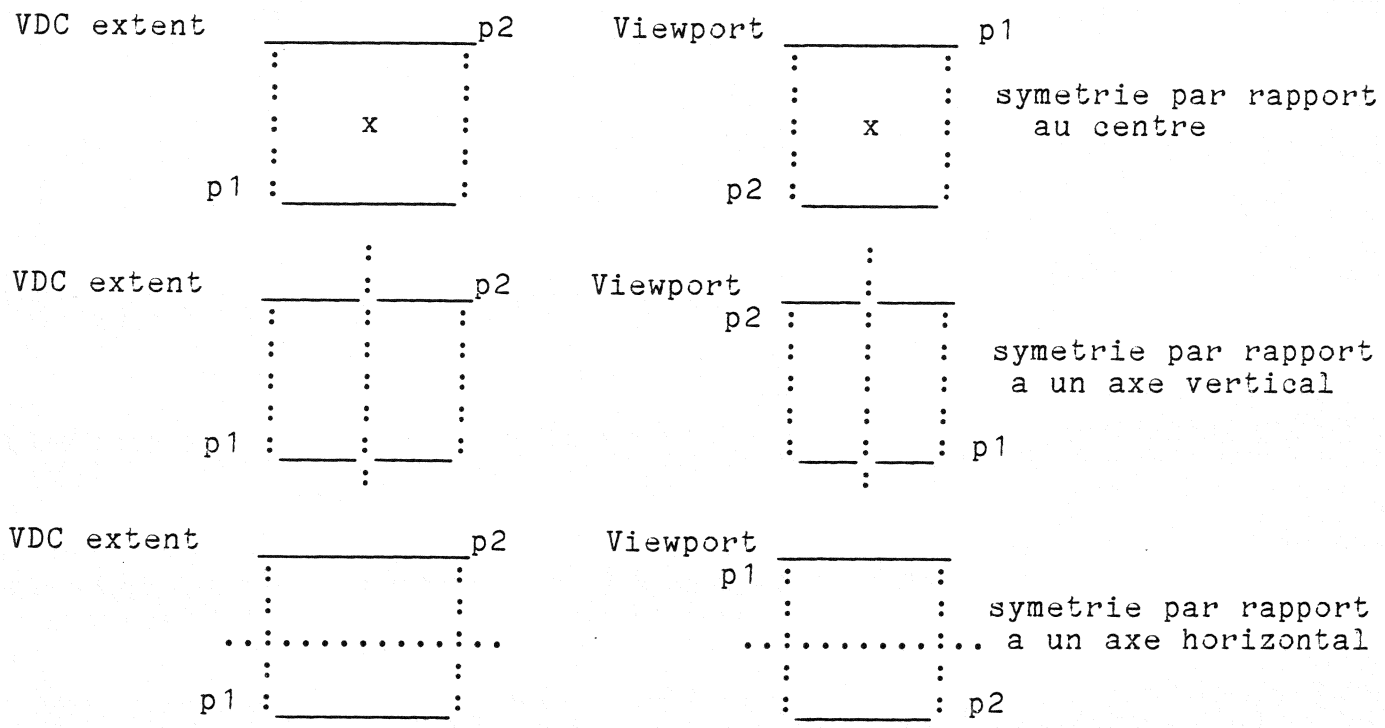
Autre note : rien n'empeche l'utilisateur de faire un choix plus judicieux de VDC extent, pour eviter des coupes aussi sombres dans sa fenetre de visualisation.

3.4.2 Effets de miroirs

L'ordre dans lequel sont specifies les points definissant VDC extent et le viewport est important : le premier point de VDC extent (resp. le deuxieme) sera mappe sur le premier point du viewport (resp. le deuxieme).

Ceci permet de visualiser des images en les transformant selon des symetries (par rapport a des droites paralleles aux axes, ou par rapport au point central des espaces) :

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse



3.5 Orientation spatiale et angulaire dans VDC extent

Un certain nombre de routines font appel aux notions droite, gauche, haut et bas (orientation d'un texte), ou a celle de sens de rotation (arc de cercle par exemple). Dans VDC space ces notions ont leur signification habituelle (sens de rotation positif, des x positifs vers les y positifs). Dans le viewport, elles subissent les effets de miroirs dus au mapping : attention a l'ordre de definition des points extremes de VDC extent et du viewport.

Nous verrons plus loin que CGI offre un autre moyen de bouleverser ces notions a l'interieur meme de VDC extent (pour l'ecriture de texte seulement).

3.6 Deplacements de VDC extent et du Viewport

Pour CGI, la situation est claire : toute modification de VDC extent ou du viewport entraine une modification du mapping. Dans un environnement VSDD ou la notion de viewport n'est plus aussi simple, il faudra reviser quelque peu cette regle.

3.6.1 Modification de VDC extent (Viewport fixe)

Le viewport est fixe ---> la surface de visualisation va restee inchangee (en taille et en position). Le contenu va etre

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

modifie en fonction du type de modification demande.

3.6.1.1 Translation de VDC extent

VDC extent se deplace dans l'espace virtuel, sans changer de taille : les facteurs d'echelle du mapping ne sont pas modifies, le contenu du viewport subit lui aussi une translation (scroll). La mise a jour de la bande liberee (cachee auparavant) peut se faire au niveau de CGI si le travail execute precedemment a été conserve, ou au niveau de l'application.

L'effet produit revient a deplacer sous le viewport un espace de travail plus grand.

3.6.1.2 Modification de la taille de VDC extent

Une zone plus petite (ou plus grande) va etre visualisee sur le meme espace de visualisation. Les facteurs d'echelle du mapping vont etre modifies ---> effet de zoom (ou de grand angle).

Ici aussi, CGI a besoin de retrouver le travail precedemment realise.

Ces 2 types de modifications de VDC extent pourront bien sur etre combines.

3.6.2 Modification du viewport, VDC fixe

Dans un environnement general, le mapping sera recalcule, un nouvel espace de travail sera defini.

Dans notre environnement VSDD, tout deplacement du viewport dans le bitmap initial, va provoquer un changement de mapping (translation simple, ou accompagnee d'un changement d'echelle), suivant le type de deplacement.

La particularite de cet environnement est d'avoir une surface disponible mouvante : un objet VSDD peut se deplacer dans l'ecran, et changer de taille.

3.6.2.1 Translation de l'objet

C'est a la charge de VIDEOMAN et du VSDD. Cette transformation est completement transparente pour CGI puisque l'objet conserve le meme systeme de coordonees locales.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

3.6.2.2 Modification de la taille de l'objet

CGI devra en etre informe. 2 reactions sont possibles :

- redefinir le mapping : la meme image virtuelle sera visualisee dans un espace plus grand ou plus petit ----> effet de loupe.

- conserver le meme mapping et visualiser plus d'informations (ou moins si la fenetre a retreci). Pour CGI, conserver le mapping revient a modifier VDC extent, pour assurer la continuite des fonctions de mapping.

Les 2 solutions seront vraisemblablement implantees.

3.6.3 Memorisation des fonctions CGI

Deux modes de fonctionnement de CGI seront proposes :

- aucune memorisation n'est realisee : tout ce qui sera trace en dehors de VDC extent sera perdu. La regeneration de l'image lors des scrolls et des changements d'echelle sera a la charge de l'application.

- CGI memorisera les appels de fonctions et sera donc en mesure de regenerer l'image quelle que soit la situation. Le format de memorisation sera probablement le format binaire CGM (voir le chapitre consacre a ce sujet).

3.7 Autres elements de controle de l'environnement

3.7.1 Controle de la session

CGI dispose de 3 fonctions permettant d'initialiser une session de travail, de la fermer et d'effectuer un reset sur tous les attributs qui prennent alors les valeurs par defaut proposees par CGI.

Extensions :

- A priori, CGI travaille sur un seul terminal a un instant donne. Toutes les implementations rencontrees ont etendu la notion d'initialisation de session, en introduisant un parametre, que l'on retrouvera dans toutes les fonctions appelees : l'identification du peripherique utilise. Plusieurs sessions pourront donc etre ouvertes en meme temps, chacune disposant de son propre environnement.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- Une autre extension frequemment rencontree est une fonction d'ouverture avec une liste de valeurs initiales pour tous les attributs (la fonction proposee par CGI initialise tous les attributs par defaut, et exige une serie d'appels aux fonctions d'initialisation des attributs). En corollaire, une fonction de reset aux valeurs d'initialisation sera proposee.

3.7.2 Controle de l'image

2 fonctions permettent un clear de la surface de visualisation (avec la couleur de fond du terminal), et la visualisation a la demande, de l'image en cours de construction (ceci pour les peripheriques qui ne peuvent pas mettre a jour leur surface de visualisation dynamiquement).

Enfin une fonction optionnelle permet de modifier la couleur de fond utilisee.

3.7.3 Clipping

Un rectangle de clipping peut etre specifie dans VDC extent. Il sera utilise suivant la valeur d'un indicateur qui peut prendre les 3 valeurs suivantes :

- "clip rectangle" : le clipping est realise sur l'intersection du rectangle de clipping et du viewport effectif.
- "viewport" : seul le viewport effectif definit le clipping. Le rectangle de clipping est ignore.
- "none" : pas de clipping realise a partir de CGI. L'application le prend en charge.

3.7.4 Controle des couleurs

CGI gere indifferemment des peripheriques couleur ou monochromes. L'application reference les couleurs par leur index dans une table qu'elle peut elle-meme initialiser, ou par leur valeur absolue RGB (3 valeurs entieres de 0 a 255).

Cas d'un peripherique couleur :

CGI connait les caracteristiques du terminal, et peut limiter la taille de la table des couleurs, ou corriger les valeurs absolues donnees, en fonction de la resolution couleur et des couleurs supportees.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

Cas d'un peripherique monochrome :

Deux couleurs sont connues : la couleur de fond, et la couleur d'ecriture. CGI peut cependant generer des echelles de gris pour les peripheriques qui les supportent.

3.7.5 Escape

Cette routine permet a une implementation CGI de proposer un ensemble de fonctions de controle non standardisees. Elle accepte en entree un numero de fonction et un buffer de parametres specifiques. Ces fonctions peuvent etre liees a un peripherique particulier, ou avoir une portee plus generale.

3.7.6 Fonctions diverses

La plupart de ces fonctions ne seront pas implementees (sauf demande express)

- definition du type de coordonnees VDC : entieres ou reelles
- definition des precisions demandees pour chaque type d'attribut : taille des attributs reels, des attributs entiers, des attributs de type indice, des coordonnees reelles ou entieres ... Ces parametres seront fixes par l'implementation.
- interrogation du terminal : permet de connaitre les types de coordonnees supportes et leurs precisions, les caracteristiques physiques du terminal... L'interet de ces fonctions est tres reduit si les precedentes ne sont pas supportees.
- interrogation de CGI : pour connaitre les valeurs courantes des attributs (ces fonctions pourront etre implantees).
- recuperation des erreurs : CGI peut gerer une pile des erreurs rencontrees, et fournit quelques points d'entree pour y acceder : pop, clear et status.

4 LES FONCTIONS GRAPHIQUES

Elles se repartissent dans 3 sous-ensembles :

- les fonctions de dessin, qui decrivent les traces.
- les fonctions d'attributs, qui positionnent les attributs de dessin (couleurs, types de traces...)

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- les fonctions de controle, qui modifient la signification de certains attributs.

Un quatrieme groupe de fonctions englobe les fonctions d'interrogation de CGI et du peripherique.

4.1 Les fonctions de dessin et les attributs associes

Ces fonctions decrivent les traces a realiser dans VDC space. Chaque appel de fonction est totalement independant des appels precedents : la notion de position courante et de deplacement relatif n'existe pas sous CGI (en accord avec le modele GKS), ceci afin d'eviter d'eventuels cumuls d'incertitudes (arrondis...), et de minimiser les risques d'effets de bord.

Il y a 4 types de fonctions de dessin, chaque type travaillant avec un groupe d'attributs specifiques :

- les traces lignes ----> attributs lignes (couleur, type et taille).
- les traces markers ----> attributs markers (couleur, type et taille).
- les traces texte ----> attributs texte (couleur, taille, orientation...).
- les traces de remplissage ----> attributs perimetres pour delimiter les figures dessinees, et attributs de remplissage pour les remplir.

..1.1 traces lignes

- polyline et disjoint polyline, sur une liste de points VDC. Dans le premier cas, tous les segments reliant les points seront traces, dans le deuxieme cas, un segment sur 2 sera ignore.
- arcs de cercles : 2 moyens de les specifier, dont l'utilisation depend de l'objectif recherche :
  + par 2 points definissant les extremites de l'arc et un point intermediaire : l'exactitude est assuree pour les extremites, mais il peut y avoir une incertitude sur la calcul du centre; utile pour relier un arc a des points precis.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

+ par le centre et le rayon du cercle, et deux vecteurs definissant le debut et la fin de l'arc. Ici, l'incertitude se porte sur les extremites de l'arc.

- arcs d'ellipses : specifies par l'ellipse et 2 vecteurs definissant le debut et la fin de l'arc. L'ellipse peut etre orientee de facon quelconque dans l'espace, et est definie par 2 diametres : l'ellipse consideree est celle dont les tangentes aux extremites d'un diametre, sont paralleles a l'autre diametre (!). Donner le grand et le petit diametre est la facon la plus claire de specifier l'ellipse, mais ce n'est qu'un cas particulier de cette definition.

Le sens de trace des arcs se fait a partir du vecteur de debut, suivant le sens de rotation positif, jusqu'au vecteur de fin.

A noter qu'il n'existe pas de fonctions lignes pour les cercles et les ellipses. Il faut definir des arcs dont les vecteurs de debut et de fin coincident.

Attributs associes a ces traces :

- type de trace : 5 types sont exiges par CGI, mais toute implementation peut en proposer d'autres :

  + ligne continue _____

  + ligne pointillee .....

  + ligne tiret - - - - - - - - - - - -

  + ligne tiret-point -.-.-.-.-.-.

  + ligne tiret-point-point -..-..-..-..

- couleur : couleur utilisee pour les traces.

- largeur : CGI gere obligatoirement cet attribut, mais peut se contenter de proposer une liste des valeurs supportees (eventuellement reduite a 1 seule largeur). Deux modes de specification sont possibles :

  + en unite VDC : la largeur de ligne subit la transformation de mapping. (on assure ainsi une proportion constante entre la largeur des lignes et les dessins).

  + par un facteur d'echelle lie au peripherique : ce facteur s'appliquera a une valeur de base liee au peripherique, par exemple la plus petite unite de visualisation : sur un moniteur, un facteur de 5 donnera une epaisseur de ligne de 5 pixels, quel que soit le mapping utilise.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

Les largeurs de ligne a destination de l'ecran, pour la machine 16 bits varieront de 1 a 16 pixels.

4.1.2 traces markers

Une seule fonction entre dans cette categorie : elle place des "markers" sur chaque point d'une liste VDC. Les markers sont associes aux attributs suivants :

- type : 5 types sont exiges par CGI, mais une implementation peut en proposer d'autres:
  - + plus +
  - + asterisque *
  - + cercle o
  - + point .
  - + croix x
- couleur : couleur des markers.
- taille : memes remarques que pour les lignes , a savoir :
  - + CGI peut ne gerer qu'une liste finie de valeurs, eventuellement reduite a 1 taille.
  - + 2 modes de specification : unites VDC et facteur d'echelle.
  - + tailles implementees pour un ecran : 1 a 16 pixels.

Le type point (.) est invariable en taille.

4.1.3 traces texte

3 fonctions permettent d'ecrire des chaines de caracteres (la notion de caractere simple n'existe pas). Le principe est le suivant :

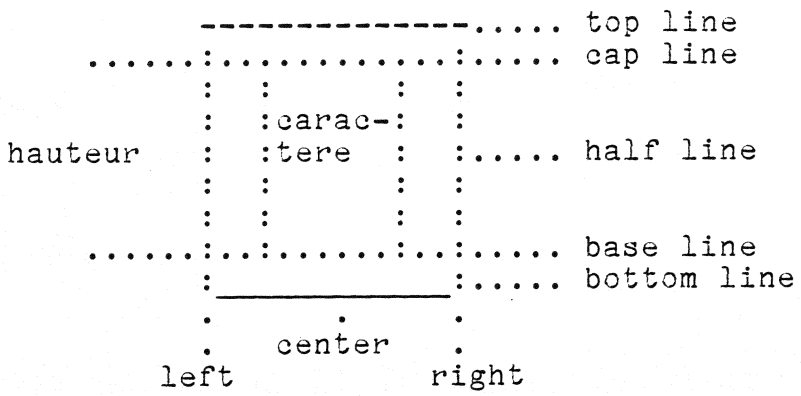
CGI autorise le decoupage d'une chaine de caracteres en plusieurs troncons, afin de permettre une modification des attributs texte entre chaque troncon. Des qu'une chaine est terminee, CGI l'englobe dans un parallelogramme minimum, sur lequel sont definis des points de repere utilises pour positionner le texte par rapport a un point de reference donne (voir attribut d'alignement).

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

L'affichage du premier tronçon d'un texte est realise par la routine TEXT, qui precise le point de reference utilise pour toute la chaine. Les tronçons suivants sont specifies par la routine APPEND TEXT, jusqu'au tronçon final identifie par un flag.

Une routine similaire a TEXT permet de fixer le parallelogramme englobant : CGI ajustera les attributs de facon a obtenir le resultat voulu.

Definition des lignes de repere sur les caracteres :



Dans le cas d'ecriture horizontale, les caracteres sont alignes selon leur baseline respective. Dans le cas d'ecriture verticale, c'est la ligne du milieu qui sert de repere d'alignement.

Attributs texte :

- couleur : couleur de la tete d'ecriture.
- orientation : specifie 2 vecteurs d'orientation du texte (horizontal et vertical). Ces 2 vecteurs peuvent etre orientes de facon quelconque et ne sont pas forcemment orthogonaux. Ils redefinissent pour les fonctions de texte la signification des 4 directions :
  - + droite = direction du vecteur horizontal.
  - + gauche = direction opposee a droite.
  - + haut = direction du vecteur vertical.
  - + bas = direction opposee a haut.

Par defaut, CGI considere que le vecteur d'orientation horizontale est defini par (0,0) et (0,1), et le vecteur d'orientation verticale par (0,0) et (1,0) : les 4 directions gardent ainsi leur signification habituelle.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- direction : definit le sens d'ecriture (une des 4 directions specifiees plus haut). Par defaut CGI ecrit vers la droite.

- hauteur : distance entre capline et baseline, mesuree le long du vecteur d'orientation verticale. (distance VDC ---> subit la transformation de mapping).

- facteur d'expansion : facteur applique sur la hauteur des caracteres, pour definir leur largeur (en plus du facteur lie a la fonte utilisee). La largeur d'un caractere sera definie par le produit des valeurs suivantes :

  + la hauteur du caractere

  + le facteur largeur/hauteur donne par la fonte (defini pour toute la fonte ou pour chaque caracterere.

  + le facteur d'expansion.

  + le rapport des longueurs des vecteurs d'orientation.

- l'espacement des caracteres : espacement laisse entre 2 caracteres. C'est un facteur reel applique sur la hauteur. Par defaut, il vaut 0 : les matrices qui contiennent les caracteres sont adjacentes. Il peut prendre une valeur negative, permettant ainsi le chevauchement de matrices.

- alignement : definit le positionnement du parallelogramme englobant la chaine terminee. 3 reperes verticaux , et 5 reperes horizontaux sont definis sur ce parallelogramme :

  Sens d'ecriture horizontal :

  + topline = la plus haute des toplines.

  + capline = la plus haute des caplines.

  + bottomline = la plus basse des bottomlines.

  + baseline = ligne d'alignement commune a toutes les matrices.

  + halfline = ligne situee a egale distance de topline et bottomline.

  + left = limite gauche du caractere le plus a gauche.

  + right = limite droite du caractere le plus a droite.

  + center = ligne a egale distance de left et right.

  Sens d'ecriture vertical:

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

+ topline = topline du caractere superieur.

+ capline = capline du caractere superieur.

+ bottomline = bottomline du caractere inferieur.

+ baseline = baseline du caractere inferieur.

+ halfline = ligne situee a egale distance de topline et bottomline.

+ left = limite gauche la plus a gauche.

+ right = limite droite la plus a droite.

+ center = ligne d'alignement commune a toutes les matrices.

L'alignement du parallelogramme est ensuite realise en positionnant l' intersection d'une ligne repere horizontale et d'une ligne repere verticale sur le point de reference donne :

+ alignement horizontal = left ou center ou right.

+ alignement vertical = topline ou capline ou halfline ou baseline ou bottomline.

2 autres types d'alignement peuvent etre specifies : normal et continu

+ normal = valeurs predefinies par CGI, dependant du sens d'ecriture.

+ continu = requiert un parametre reel supplementaire pour definir la distance a placer entre le parallelogramme et le point de reference. Ce parametre est un facteur a multiplier par la dimension du parallelogramme pour obtenir la distance voulue. (horizontale ou verticale suivant l'alignement considere).

- precision des clipping texte : 3 precisions possibles :

+ chaine : les chaines ne sont pas clippees. Une chaine sera visible si le debut de la chaine est a l'interieur du rectangle de clipping.

+ caractere : les chaines sont clippees, mais pas les caracteres.

+ point : on clippe meme a l'interieur d'un caractere. (option supportee sur la machine 16 bits).

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- fonte : indique la fonte courante parmi la liste des fontes disponibles. CGI n'impose aucune directive concernant la gestion des fontes.

- jeux de caracteres (ref = norme ISO 2022) : permettent l'acces a la fonte en definissant le type de jeux de caracteres utilises (7 bits ou 8 bits , ou eventuellement jeu non standard), et le mode d'extension de ces jeux. Deux jeux comprenant les codes hexa 20 a 7f sont accessibles en meme temps, l'acces au deuxieme jeu se faisant sur le bit 7 (jeux 8 bits), ou en utilisant les caracteres de controle SI et SO (jeux 7 bits). L'acces a la fonte peut etre redefini par l'implementation, la seule exigence etant de supporter le jeu de caracteres ASCII dans une fonte quelconque.

4.1.4 traces remplissages

Les fonctions de ce groupe utilisent 2 types d'attributs : les attributs perimetres qui delimitent les figures demandees, et les attributs de remplissage qui permettent de representer l'interieur de ces figures.

-----> polygones : specifies comme une polyline ; un segment supplementaire sera trace entre le dernier point et le premier. Il n'y a aucune restriction sur l'aspect du polygone : en particulier, des segments peuvent se croiser. Une deuxieme routine offre 2 extensions :

- plusieurs polygones peuvent etre definis en meme temps (flags de fin de polygone dans la liste de points).

- la visibilite de chaque segment est parametrable.

Enfin, le cas particulier des rectangles paralleles aux axes a ete considere : une routine s'en charge et necessite seulement 2 parametres : 2 points definissant une diagonale. opposes.

-----> cercles et ellipses : ce sont ici des figures remplies. Un cercle est defini par son centre et son rayon, et l'ellipse par 2 diametres.

Les arcs de cercle et les arcs d'ellipse fermes sont possibles. 2 types de fermeture sont proposes : pie (2 segments reliant les extremites des arcs au centre) et chord (corde entre les 2 extremites).

Attributs perimetres :

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

Ils ont les memes specifications que les attributs lignes (couleur, type et largeur). Un attribut specifique aux perimetres a ete introduit :

- visibite des frontieres : on ou off. Les fonctions peuvent devenir uniquement des fonctions de remplissage, sans marquage des limites.

Attributs de remplissage :

- couleur : couleur de remplissage.
- style : definit le mode de remplissage :
  + vide : les fonctions deviennent des fonctions ligne, avec les attributs perimetres (a condition que la visibilite des frontieres soit demandee !)
  + plein : la figure est remplie de facon uniforme avec la couleur de remplissage.
  + creux: une epaisseur de couleur de remplissage delimite l'interieur de la figure (pas de valeur definie par CGI).
  + hachures : la figure est remplie par des hachures. L'utilisation d'un point de reference commun pour commencer les calculs, assure la continuite des hachures dans 2 figures adjacentes.
  + pattern : remplissage avec un pattern. Le point de reference commun est utilise.
- type de hachures : 6 types possibles :
  + verticales.
  + horizontales.
  + obliques (angle +45 degres).
  + obliques (angle -45 degres).
  + verticales et horizontales.
  + obliques dans les 2 sens.
- pattern utilise : index dans la table des patterns utilisateurs
- taille du pattern : dimension VDC du pattern.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- point de reference : point de depart commun pour tous les calculs de positionnement des hachures et des patterns.

4.1.5 La fonction CELL ARRAY

C'est une fonction de dessin qui n'entre dans aucune des categories deja citees. Elle permet de definir un parallelogramme quelconque dans l'espace virtuel, et de le decouper en nx tranches dans une dimension, et en ny tranches dans l'autre dimension. On definit ainsi nx * ny cellules, qui vont etre coloriees suivant une liste de couleurs fournie en parametre.

4.1.6 La fonction GDP

Generalized Drawing Primitive : c'est l'equivalent de la fonction ESCAPE, pour les fonctions de dessin. Toute fonction non standardisee supportee par une implementation doit etre appelee a travers GDP. Outre l'identificateur de la fonction appelee, cette fonction recoit en parametres une liste de points VDC et un buffer de parametres specifiques. Elle dispose de tous les attributs courants de CGI.

4.2 Fonctions de controle

4.2.1 Modes de specification

Les fonctions de ce groupe precisent les modes de specification de quelques attributs , et definissent a CGI l'interpretation qui doit en etre faite :

- specification des couleurs : par index dans une table ou par valeur absolue RGB.

- specification des epaisseurs de lignes, perimetres et markers : en unite VDC ou comme facteur d'echelle lie au peripherique.

4.2.2 Gestion des couleurs

- definition de la table des couleurs utilisees (pour un mode de specification par index).

- definition du mode transparence : par defaut le mode transparence est positionne : l'image n'est pas ecrasee derriere les matrices de caracteres, derriere les "trous" des

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

lignes discontinues ... Ce mode de fonctionnement peut etre annule. La couleur "transparente" est alors remplacee par la couleur auxiliaire specifiee.

- definition de la couleur auxiliaire : en mode transparence off.

4.2.3 ASF

Jusqu'a present, nous avons parle des attributs de dessin sans nous preoccuper de leur existence materielle. Ces attributs sont classes selon le critere suivant :

Attributs individuels : ils sont representes par une variable unique, contenant leur valeur courante. (gestion classique)

Attributs groupes : CGI dispose de 5 tables de valeurs predefinies, associees aux 5 categories d'attributs :

- attributs lignes.
- attributs markers.
- attributs texte.
- attributs perimetres.
- attributs de remplissage.

Une entree dans une table est une configuration proposee pour les attributs associes. Par exemple, 2 entrees dans la table des attributs lignes pourront etre :

ligne rouge continue, d'epaisseur 5,

- ligne pointillee bleue, d'epaisseur 1.

5 fonctions permettent de definir l'entree courante dans chaque table, et par consequent la configuration courante utilisee. Les attributs groupes conservent neanmoins leur valeur courante individuelle definie par les fonctions vues plus haut. Un tableau de flags permet a CGI de selecter la valeur individuelle de l'attribut, ou la valeur prise dans la configuration courante.

Ce tableau de flags est accessible a travers la routine ASF (Aspect Source Flags).

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

Liste des attributs groupes :

- attributs lignes : type, couleur et taille.
- attributs markers : type, couleur et taille.
- attributs perimetres : type , couleur et taille.
- attributs de remplissage : style, couleur, hachures et pattern.
- attributs texte : fonte, couleur, precision, espacement, et expansion.

Tous les autres attributs sont individuels.

5 OPTIONS SETS

5.1 Figures fermees

Cette option permet de construire des figures comme combinaison de fonctions lignes et GDP. La definition d'une telle figure est encadree par 2 appels de fonctions (BEGIN FIGURE et END FIGURE) et ne peut contenir que des appels a des fonctions lignes, GDP ou attributs : le perimetre construit sera une suite de segments et d'arcs.

Notes :

- Les attributs perimetres remplacent ici les attributs lignes.
- CGI construit implicitement les segments qui relient le dernier point de chaque fonction au premier de la suivante.
- La visibilite de ces segments est parametrable.
- A tout moment de la construction, une fonction CLOSE peur etre appelee. Elle ferme la figure (mais ne termine pas la definition).
- A la fin de la definition, un CLOSE implicite est realise, et la figure est remplie en fonction des attributs de remplissage.

5.2 Configurations d'attributs utilisateurs

Nous avons vu que CGI sait gerer des tables de configurations d'attributs. Le noyau minimum de CGI prevoit seulement l'existence de ces tables, et leur pre-initialisation avec des configurations supportees par l'implementation.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

Un ensemble de fonctions optionnelles autorise l'application utilisatrice a creer ses propres configurations : les tables deviennent dynamiques. Pour chaque famille d'attributs, une fonction permet d'initialiser une entree de la table associee, avec une serie de valeurs.

5.3 Segmentation

Un segment est un ensemble de fonctions CGI, identifie par un nom, et utilisable comme un entite simple. Cette fonctionnalite tente de definir une notion d'objet graphique, comme une combinaison d'objets elementaires connus.

5.3.1 Creation d'un segment (ouverture)

Un identificateur lui est assigne (par l'utilisateur ou par CGI), et les valeurs courantes des attributs de dessin lui sont associees. Il est important de noter que ces valeurs sont definitivement associees au segment. Elles ne pourront etre modifiees que par un appel explicite d'une fonction attribut a l'interieur du segment.

5.3.2 Attributs d'un segment

- les attributs de dessin copies au moment de la creation.
- une priorite d'affichage permettant de gerer les recouvrements de segments.
- des attributs de visibilite et d'aspect (surbrillance, video inverse ...)
- une matrice de transformation permettant d'effectuer translations, rotations et scalings.

Seuls les attributs de dessin sont inaccessibles depuis l'exterieur du segment. Pour les autres, des fonctions d'acces permettent de les modifier a tout moment, en dehors de la definition du segment.

5.3.3 Elements d'un segment

Toutes les fonctions de dessin et d'attributs sont admises comme elements de segment, ansi que quelques fonctions de controle. D'autres fonctions sont autorisees a figurer dans la definition du segment, mais ne sont pas memorisees comme elements

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

(la modification de la table des couleurs par exemple), d'autres enfin sont interdites (deplacer VDC extent).

Au cours de la definition, toutes les fonctions appelees sont executees, visualisant ainsi la construction.

A la fin de la definition (fermeture), la liste des elements est cataloguee sous le nom du segment, et devient utilisable comme element de CGI.

5.3.4 Utilisation d'un segment

A chaque utilisation d'un segment, une matrice de transformation est appliquee sur les points VDC, parametres des fonctions elements. Par defaut, cette matrice est definie comme la matrice identite.

CGI applique cette transformation sur les points VDC uniquement, ce qui reduit les possibilites et peut conduire a des surprises :

- un cercle defini par son centre et son rayon pourra etre translate, mais ne pourra pas changer d'echelle. Il faudra le definir par 3 points.

- attention egalement aux specifications des fonctions CGI : en appliquant une matrice de rotation aux 2 coins qui definissent un rectangle, on ne fait pas subir une rotation au rectangle dans son ensemble : les 2 points sont transformes, puis la fonction CGI s'applique ----> le rectangle est trace parallelement aux axes !!

5.3.5 Operations sur les segments

- copie dans un autre segment : la copie se fait en appliquant une matrice de transformation sur le segment copie (la matrice attribut du segment est ignoree). C'est une copie element par element, ce qui veut dire :

  + que les attributs associes au segment copie disparaissent, au profit des attributs du segment pere.

  + que les elements copies ne sont plus associes au segment d'ou ils sont issus : les modifications ulterieures de ce segment ne les affecteront pas.

- affichage : d'un segment, ou de tous les segments definis, par ordre de priorite.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- destruction : d'un segment ou de tous les segments definis.

- changement du nom d'un segment.

- modification d'un segment : des elements peuvent etre ajoutes a la fin d'un segment.

- matrice de transformation : une matrice attribut est associee au segment.

- modification des attributs de visibilite et de priorite.

Remarque : ces specifications n'autorisent pas l'utilisation de parametres dans les segments. Le seul moyen de transformer un objet rouge en objet bleu va etre la copie de segments :

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

```
   objet rouge                    objet bleu

Fill Colour (red)              Fill Colour (blue)
Open Segment (seg_1)           Open Segment (seg_2)
     :                              Copy Segment (seg_1,identity_matrice)
     :                         Close Segment
     :
Close Segment
```

Cette facon de proceder alourdit considerablement les softs. On pourra envisager une extension de CGI, permettant une utilisation plus souple.

5.4 Bitmaps

Un dernier ensemble optionnel permet l'utilisation de bitmaps comme peripheriques de sortie.

5.4.1 Creation des bitmaps

2 formats sont possibles :

- format respectant le format du peripherique reel associe (resolution couleur...).
- format 1 bit/pixel.

Plusieurs bitmaps peuvent etre crees, un seul est identifie comme le bitmap de travail, sur lequel s'appliquent toutes les fonctions CGI. Cette notion de bitmap n'empeche pas l'utilisation de coordonnees virtuelles : les transformations de mapping s'effectuent toujours.

Note : dans l'environnemnet de la machine 16 bits, ces notions existent via les bitmaps des objets geres par VIDEOMAN.

5.4.2 Operations sur les bitmaps

Outre la creation, l'affichage et la destruction d'un bitmap, CGI propose 3 fonctions operant sur les bitmaps :

Fonctions a 2 operandes :

- bitmap dest = f1 (bitmap source, bitmap dest) ---> move.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- bitmap dest = f2 (pattern, bitmap dest) ---> remplissage.

L'operation executee sur chaque bit traite est definie par un code offrant les 16 combinaisons possibles.

- code 0 : dest = 0
- code 1 : dest = source AND dest
- code 2 : dest = source AND (NOT dest)
- code 3 : dest = source
- ....
- code 9 : dest = NOT (source XOR dest)
- ...
- code 15 : dest = 1

Fonction a 3 operandes :

- Bitmap dest = f3 (bitmap dest, bitmap source, pattern)

Le code de l'operation offre 256 possibilites parmi lesquelles :

- NOT (srce XOR (ptrn AND (dest OR (NOT srce)))) !
- ptrn XOR (srce XOR (dest OR (srce AND (NOT ptrn)))) !!!
- ....

6 CGM

CGM (Computer Graphics Metafile) est une norme ISO, definissant un format standard pour stocker les fonctions graphiques qui decrivent une image.

Cette normalisation du codage d'images graphiques, assure qu'un fichier issu d'un generateur CGM sera recuperable par tout interpreteur CGM. (en supposant que les formats physiques des fichiers aient ete rendus compatibles).

Il existe une relation quasi bijective entre les fonctions connues des interpreteurs-generateurs CGM, et les fonctions CGI : plus une implementation s'eloignera de CGI, et plus la tache de ces logiciels deviendra complexe. Dans le pire des cas, la conversion peut devenir impossible si les fonctionnalites supportees sont trop specifiques.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

6.1 Elements propres a CGM

Ces elements ne sont pas des fonctions executables par CGI. Ils sont destines aux interpreteurs CGM, et figurent dans les fichiers en tant qu'elements de controle (BEGIN PICTURE, END PICTURE ...).

6.2 Elements de CGI inconnus dans CGM

- Toutes les fonctions decrivant l'environnement reel : le transport sur fichier ne concerne evidemment que l'image virtuelle : une application doit definir son environnement reel avant d'interpreter un fichier CGM.

- Certains "option sets", et en particulier toutes les fonctions relatives a la segmentation. Les segments peuvent cependant etre encodes sous CGM, element par element : ils perdront leur statut d'entite graphique. Ce ne sera pas tres grave dans le cas d'un transport simple d'image, ca le deviendra si l'image transportee est susceptible d'etre modifiee (possible dans le cas de fichiers CGM codes texte).

6.3 Codages CGM

3 codages sont possibles, chacun repondant a un objectif different :

- Codage texte :

  Les fonctions et leurs parametres sont codes en ASCII, le resultat etant un fichier texte editable. La taille des fichiers generes est donc relativement importante, et les logiciels d'interpretation doivent supporter des analyseurs syntaxique et semantique.

- Codage caracteres :

  C'est le plus compact en ce qui concerne la taille des fichiers generes. Toutes les fonctions et leurs parametres sont codes numeriquement, de facon a occuper le moins de place possible : en particulier le codage des valeurs numeriques utilise un format variable (il faut moins d'octets pour representer la valeur 1 que la valeur 1000). Le code utilise est un code 7 bits, ce qui justifie le choix de ce codage dans les environnements qui ne supportent que ce type de transmissions.

  Dans tous les octets, un bit differencie les fonctions des parametres, et un autre bit sert de flag d'extension . Il reste donc 5 bits pour encoder les valeurs. Pour les entiers,

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

1 bit de signe est present, ce qui laisse 4 bits de valeur pour chaque octet !!!

Exemple : 1 est code $41 (bit parametre et valeur).
-1 est code $51 (+ bit signe : pas de representation en complement a 2)
un octet couvre les valeurs -15 a 15
16 exige 2 octets : $6050
deux octets couvrent -255 a +255

Les valeurs entieres manipulees par CGI sont le plus souvent des selections d'attributs (valeurs inferieures a 16), ou des points VDC (les applications travaillent souvent autour du point (0,0), et utilisent rarement toute la largeur disponible ----> le gain de place est assure par rapport a un format fixe.

Les generateurs-interpreteurs ne sont par contre pas minimums pour ce genre de codage puisqu'ils doivent assurer toutes les conversions.

- Codage binaire

C'est le codage qui demande le moins de travail aux generateurs -interpreteurs. Toutes les valeurs numeriques conservent leur format machine (en 8,16 ou 32 bits, 64 bits pour les flottants (norme ISO)). Ce codage requiert cependant des possibilites de transmissions sur 8 bits significatifs, dans le cas de transfert de fichiers.

6.4 Implementation de CGM

Toutes les implementations proposent un codage binaire CGM, et pour la plupart un codage texte, qui peut varier par rapport au standard. En particulier MATRA HARRIS sur son VDS (VSDD development system), propose la generation de fichiers sources C, a partir de la definition interactive d'une image. Ces fichiers peuvent ensuite etre modifies (pour y inserer des structures de controle par exemple, les fichiers generes etant strictement sequentiels), et deroules par un interpreteur C (on evite ainsi les phases de compilation et de link lors de la mise au point d'une image).

Sur la machine 16 bits, le codage binaire sera de toute facon necessaire a CGI pour memoriser les fonctions appelees de facon a regenerer les parties cachees d'une image. Rien n'est encore decide en ce qui concerne les autres codages proposes.

7 CONCLUSION

CGI offre donc les moyens de dialoguer de facon standard avec des peripheriques quelconques (utile pour naviguer parmi la

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

diversite de la gamme d'imprimantes du marche), et associe a CGM permet le transfert d'image developpees sur moniteur, dont la destination finale sera un terminal papier quelconque.

Cependant plusieurs limites ont ete entrevues ; elles necessiteront un certain nombre d'ouvertures hors standard. L'utilisation de ESCAPE et GDP autorise toutes les extensions, mais diminuera d'autant la portabilite sur d'autres systemes (est-ce primordial, ou se contente-t-on de transporter des images a l'interieur de notre seul systeme, sur un ensemble de peripheriques ?).

8 LISTE DES FONCTIONS PREVUES DANS L'IMPLEMENTATION

Les fonctions soulignees font partie du noyau minimum.

8.1 Fonctions de Controle

- INITIATE = initialisation d'une session CGI (2 versions seront proposees : initialisation avec les valeurs par defaut de CGI, et initialisation avec une table de valeurs proposees par l'utilisateur).
- RESET = reset des attributs (valeurs par defaut ou valeurs d'initialisation.
- TERMINATE = fin de session.
- VDC EXTENT = definition de VDC extent.
- DEVICE VIEWPORT = definition du viewport.
- DEVICE VIEWPORT SPECIFICATION UNIT = definition des unites utilisees pour specifier le viewport (pourcentage, 1/10 mm, coordonnees physiques).
- DEVICE VIEWPORT SPECIFICATION MAPPING = definition du mapping (isotropie forcee ou pas, comment construire le viewport effectif ?)
- CLIP RECTANGLE = definition du rectangle de clipping dans VDC extent.
- CLIP INDICATOR = interpretation du clipping (clip rectangle, viewport, none).
- CLEAR VIEW SURFACE = remplit la surface de visualisation avec la couleur de fond.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- MAKE PICTURE CURRENT = visualisation de l'image construite.
- BACKGROUND COLOR = definition de la couleur de fond.
- ESCAPE = appel d'un fonction de controle non standard.

8.2 Fonctions graphiques

8.2.1 Fonctions qui utilisent les attributs lignes

- POLYLINE = dessine une polyline.
- DISJOINT POLYLINE = dessine une serie de segments disjoints.
- CIRCULAR ARC 3 POINTS = arc de cercle defini par 3 points.
- CIRCULAR ARC CENTER = arc de cercle defini par centre, rayon, debut et fin d'arc.
- ELLIPTICAL ARC = arc d'ellipse defini par le centre et 2 diametres de l'ellipse, et debut et fin d'arc.

8.2.2 Fonctions utilisant les attributs markers

- POLYMARKER = place un marker sur chaque point VDC specifie.

.2.3 Fonctions utilisant les attributs texte

- TEXT = dessine une chaine de caracteres, a partir d'un point de reference.
- RESTRICTED TEXT = meme chose, en forcant la chaine a se trouver dans un parallelogramme specifie.
- APPEND TEXT = suite de la chaine de caractere.

8.2.4 Fonctions utilisant les attributs perimetres et remplissage

Toutes ces fonctions remplissent la figure dessinee.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- CIRCLE = cercle defini par centre et rayon.
- CIRCULAR ARC 3 POINTS CLOSE = arc de cercle ferme (2 fermetures possibles : pie / chord).
- CIRCULAR ARC CENTER CLOSE = meme chose , defini autrement.
- ELLIPTICAL ARC CLOSE = arc d'ellipse ferme (2 fermetures possibles :

  pie /chord).
- POLYGON = dessine et ferme un polygone.
- POLYGON SET = meme chose avec plusieurs polygones.

.2.5 Autres fonctions de dessin

- GDP = appel d'une fonction de dessin non standardisee. Parametres = un numero de fonction, une liste de points VDC, et un buffer de parametres specifiques.
- CELL ARRAY = division d'un parallelogramme en cellules de taille donnee, et remplissage a partir d'une liste de couleurs.

8.2.6 Attributs lignes

- LINE BUNDLE INDEX = index courant pour les attributs groupes.

  LINE TYPE = type de ligne (continue ou ..... ou _____ ou _·_·_·_· ou _··_··_··_..).
- LINE COLOR = couleur utilisee.
- LINE WIDTH = largeur (unite = facteur d'echelle de l'unite de visualisation (nb pixels) ou unite VDC)

8.2.7 Attributs markers

- MARKER BUNDLE INDEX = index courant pour les attributs groupes.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- MARKER TYPE = type des markers (* ou + ou . ou x ou o).
- MARKER COLOR = couleur utilisee.
- MARKER SIZE = taille (voir remarque ligne).

8.2.8 Attributs texte

- TEXT BUNDLE INDEX = index courant pour les attributs groupes.
- TEXT FONT INDEX = fonte utilisee.
- TEXT PRECISION = precision pour clipping (point, caractere, chaine).

  TEXT COLOR = couleur utilisee pour ecrire.
- CHARACTER EXPANSION FACTOR = facteur d'echelle hauteur/largeur.
- CHARACTER SPACING = espacement entre 2 caracteres.
- CHARACTER HEIGHT = hauteur (VDC unit).
- TEXT PATH = direction de la tete d'ecriture (haut, bas, droite, gauche).
- TEXT ALIGNMENT = positionnement du texte par rapport au point de reference.

8.2.9 Attributs perimetres

- EDGE BUNDLE INDEX = index courant pour les attributs groupes.
- EDGE TYPE = type de ligne (voir ligne).
- EDGE WIDTH = largeur (voir ligne).
- EDGE COLOR = couleur utilisee.
- EDGE VISIBILITY = flag de visualisation des frontieres des figures fermees.

8.2.10 Attributs de remplissage

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- FILL BUNDLE INDEX = index pour les attributs groupes.
- INTERIOR STYLE = definit le mode de remplissage (vide, creux, plein, hachures, pattern).
- FILL COLOR = couleur de remplissage.
- HATCH INDEX = type de hachures utilisees.
- PATTERN INDEX = type de pattern utilise (table des patterns).
- PATTERN TABLE = mise a jour de la table des patterns.
- PATTERN SIZE = definit la taille du pattern utilise.
- FILL REFERENCE POINT = definit le point de reference (assure la continuite des hachures ou des patterns).

8.2.11 Fonctions de controle

- TRANSPARENCY = on/off (si off, la couleur auxiliaire est utilisee partout ou serait utilisee la couleur transparente).
- AUXILIARY COLOR = couleur utilisee si pas de transparence.
- ASPECT SOURCE FLAGS = definit le statut de chaque attribut pouvant etre groupe (statut groupe ou statut individuel).

8.3 Option sets

Les fonctions soulignees sont les fonctions du noyau minimum de haque option set.

8.3.1 Figures fermees

- BEGIN FIGURE = initialise une figure fermee.
- END FIGURE = ferme la figure et la remplit.
- CLOSE = effectue une fermeture, mais ne termine pas la figure.
- IMPLICIT EDGE VISIBILITY = flag de visualisation des frontieres ajoutees (pour relier les differents morceaux et pour fermer).

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

8.3.2 Segmentation

- OPEN SEGMENT = definition d'un segment.
- CLOSE SEGMENT = fin definition.
- COPY SEGMENT = copie dans un autre segment.
- DELETE SEGMENT = destruction.
- DELETE ALL SEGMENTS = purge.
- RENAME SEGMENT = changement de nom.
- REDRAW SEGMENT = affichage d'un segment.
- REDRAW ALL SEGMENTS = affichage de tous les segments definis.
- SEGMENT TRANSFORM = definit une matrice de transformation.
- SEGMENT VISIBILITY = flag de visibilite.
- SEGMENT PRIORITY = priorite d'affichage.

8.3.3 Definition de configuration d'attributs

- LINE REPRESENTATION = configuration d'attributs lignes.
- MARKER REPRESENTATION = configuration d'attributs markers.
- TEXT REPRESENTATION = configuration d'attributs texte.

  EDGE REPRESENTATION = configuration d'attributs perimetres.
- FILL REPRESENTATION = configuration d'attributs de remplissage.

8.3.4 Bitmaps

- CREATE BITMAP = allocation d'un bitmap.
- DELETE BITMAP = restitution.
- SELECT DRAWING BITMAP = definit le bitmap de travail.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse

- DISPLAY BITMAP = affichage.
- DRAWING MODE = code des fonctions a 2 operandes.
- TWO OPERAND BITBLT = f1 (dest,source).
- TILE TWO OPERAND BITBLT = f2 (dest,pattern).
- TILE THREE OPERAND BITBLT = f3 (source,dest,pattern).
- DRAWING MODE 3 = code fonction a 3 operandes
- PIXEL ARRAY = init bitmap.

Propriété de THOMSON - BRANDT, ce document ne peut-être communiqué aux tiers sans autorisation expresse